

PPPPPPPPPPPP	RRRRRRRRRRRR	TTTTTTTTTTTT	SSSSSSSSSSSS	MMM	MMM	BBBBBBBBBBBB			
PPPPPPPPPPPP	RRRRRRRRRRRR	TTTTTTTTTTTT	SSSSSSSSSSSS	MMM	MMM	BBBBBBBBBBBB			
PPPPPPPPPPPP	RRRRRRRRRRRR	TTTTTTTTTTTT	SSSSSSSSSSSS	MMM	MMM	BBBBBBBBBBBB			
PPP	PPP	RRR	RRR	TTT	SSS	MMMMMM	MMMMMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMMMMM	MMMMMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMMMMM	MMMMMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPPPPPPPPPPP	RRRRRRRRRRRR	TTT	SSSSSSSSSS	MMM	MMM	BBBBB6BBBBBBB			
PPPPPPPPPPPP	RRRRRRRRRRRR	TTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBBBB			
PPPPPPPPPPPP	RRRRRRRRRRRR	TTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBBBB			
PPP	RRR	RRR	TTT	SSS	MMM	BBB	BBB		
PPP	RRR	RRR	TTT	SSS	MMM	BBB	BBB		
PPP	RRR	RRR	TTT	SSS	MMM	BBB	BBB		
PPP	RRR	RRR	TTT	SSS	MMM	BBB	BBB		
PPP	RRR	RRR	TTT	SSS	MMM	BBB	BBB		
PPP	RRR	RRR	TTT	SSS	MMM	BBB	BBB		
PPP	RRR	RRR	TTT	SSS	MMM	BBB	BBB		
PPP	RRR	RRR	TTT	SSSSSSSSSSSS	MMM	BBB	BBB		
PPP	RRR	RRR	TTT	SSSSSSSSSSSS	MMM	BBB	BBB		
PPP	RRR	RRR	TTT	SSSSSSSSSSSS	MMM	BBB	BBB		

\*\*FILE\*\*ID\*\*OUTPUT

D 11

OUT  
V04

000000	UU	UU	TTTTTTTTTT	PPPPPPPP	UU	UU	TTTTTTTTTT
000000	00	UU	UU	PP	PP	UU	UU
00	00	UU	UU	PP	PP	UU	UU
00	00	UU	UU	PP	PP	UU	UU
00	00	UU	UU	PP	PP	UU	UU
00	00	UU	UU	PPPPPPPP	UU	UU	UU
00	00	UU	UU	PPPPPPPP	UU	UU	UU
00	00	UU	UU	PP	UU	UU	UU
00	00	UU	UU	PP	UU	UU	UU
00	00	UU	UU	PP	UU	UU	UU
00	00	UU	UU	PP	UU	UU	UU
000000	UUUUUUUUUU	UUUUUUUUUU	TT	PP	UUUUUUUUUU	TT	TT
000000	UUUUUUUUUU	UUUUUUUUUU	TT	PP	UUUUUUUUUU	TT	TT

....

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	IIII	SS
LL	IIII	SS
LL	IIII	SS
LL	IIII	SSSSSS
LL	IIII	SSSSSS
LL	IIII	SS
LL	IIII	SS
LL	IIII	SS
LLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLL	IIIIII	SSSSSSSS

```

1 0001 0 MODULE OUTPUT ( %TITLE 'Print Symbiont -- output routines'
2 0002 0 IDENT = 'VO4-000'
3 0003 0 ADDRESSING_MODE (EXTERNAL = GENERAL)
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 ****
9 0009 1 *
10 0010 1 /* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 /* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 /* ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 /* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 /* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 /* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 /* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 /* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 /* TRANSFERRED.
20 0020 1 *
21 0021 1 /* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 /* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 /* CORPORATION.
24 0024 1 *
25 0025 1 /* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 /* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 ****
30 0030 1 .
31 0031 1
32 0032 1 ++
33 0033 1 /* FACILITY:
34 0034 1 Print Symbiont.
35 0035 1
36 0036 1 /* ABSTRACT:
37 0037 1 Output routines.
38 0038 1
39 0039 1 /* ENVIRONMENT:
40 0040 1 VAX/VMS user mode.
41 0041 1 --
42 0042 1
43 0043 1 /* AUTHOR: G. Robert, CREATION DATE: 31-Aug-1982
44 0044 1
45 0045 1 /* MODIFIED BY:
46 0046 1
47 0047 1 3B-006 RRB3006 Rowland R. Bradley 24-Jul-1984
48 0048 1 Add routine headers.
49 0049 1
50 0050 1 3B-005 GRR3005 Gregory R. Robert 29-Apr-1984
51 0051 1 FT2 bugfixes. Accept explicit OPEN/CLOSE calls instead
52 0052 1 of relying upon START_TASK.
53 0053 1
54 0054 1 3B-004 RRB3004 Rowland R. Bradley 27-Apr-1984
55 0055 1 Add support for storage, initialization, and ultimate
56 0056 1 display of the entire device name.
57 0057 1

```

58 0058 1 | 3B-003 GRR3003 Gregory R. Robert 23-Aug-1983  
59 0059 1 | Bugfixes, page\_setup\_modules, form\_setup\_modules,  
60 0060 1 | sheet\_feed, symbiont-initiated pause\_task and stop\_stream,  
61 0061 1 | hangup code, read and write item services  
62 0062 1 |  
63 0063 1 | 3B-002 GRR3002 Gregory R. Robert 03-Aug-1983  
64 0064 1 | Rewrite for new design.  
65 0065 1 |  
66 0066 1 | 3B-001 GRR3001 Gregory R. Robert 23-Jul-1983  
67 0067 1 | Created new module.  
68 0068 1 |  
69 0069 1 |  
70 0070 1 !\*\*

```
: 72      1 LIBRARY 'SYSSLIBRARY:LIB';
73      1 REQUIRE 'LIB$:SMBDEF';
74      1 REQUIRE 'SRC$:SMBREQ';
75      1
76      1 EXTERNAL ROUTINE
77      1          PSM$REPORT,
78      1          SMB$SEND TO JOBCTL,
79      1          PSM$STORE_ERRORS
80      1          ;
81      1
82      1 EXTERNAL
83      1          PSM$HANGUP_DISPATCH_TABLE
84      1          ;
85      1
86      1
87      1 FORWARD ROUTINE
88      1          PSM$OUTPUT
89      1          START_PRINTER    :
90      1          STOP_PRINTER    :
91      1          WRITE_AST      : NOVALUE
92      1          ;
93      1
```

: R

```
95      1040 1 %SBTTL 'PSMSOUTPUT'
96      1041 1 ! Functional Description:
97          1042 1 ?desc
98
99      1043 1 ! Formal Parameters:
100     1044 1 ?desc
101     1045 1 ! Implicit Inputs:
102     1046 1 none
103     1047 1
104     1048 1 ! Implicit Outputs:
105     1049 1 none
106     1050 1
107     1051 1 ! Returned Value:
108     1052 1 none
109     1053 1
110     1054 1 ! Side Effects:
111     1055 1 none
112     1056 1
113     1057 1 !--
114     1058 1 GLOBAL ROUTINE PSM$OUTPUT (
115     1059 1     SMB_CONTEXT : REF $LONGWORD,
116     1060 1     USER_CONTEXT : REF VECTOR,
117     1061 1     FUNCTION : REF $LONGWORD,
118     1062 1     FUNC_DESC : REF VECTOR,
119     1063 1     FUNC_ARG : REF $LONGWORD
120     1064 1     ) =
121     1065 2 BEGIN
122     1066 2
123     1067 2 SELECTONEU .FUNCTION[] OF
124     1068 2     SET
125     1069 2
126     1070 2     [PSMSK_WRITE, PSMSK_WRITE_NOFORMAT]:
127     1071 3     BEGIN
128     1072 3     LOCAL IO_FUNC : INITIAL (IOS_WRITEBLK);
129     1073 3     LOCAL SIZE, ADDRESS;
130     1074 3
131     1075 3     BIND IOB = .SMB_CONTEXT[] : $BBLOCK;
132     1076 3     BIND SCB = .IOB[IOB_A_CONTEXT] : $BBLOCK;
133     1077 3
134     1078 3     ! If device was hungup then mark it unavailable and request stop stream
135     1079 3
136     1080 3     IF TESTBITSC (SCB[PSMSV_HANGUP])
137     1081 3     THEN
138     1082 4     BEGIN
139     1083 4     SCB[PSMSL_DEVICE_STATUS] = .SCB[PSMSL_DEVICE_STATUS] OR
140     1084 4     (SMBMSGSM_UNAVAILABLE OR SMBMSGSM_STOP_STREAM);
141     1085 4     RETURN SSS_HANGUP;
142     1086 3     END;
143     1087 3
144     1088 3     IF .FUNCTION[] EQL PSMSK_WRITE_NOFORMAT
145     1089 3     THEN
146     1090 3     IF .SCB[PSMSL_DEVICE_CLASS] EQL DC$_TERM
147     1091 3     THEN
148     1092 3     IO_FUNC = IOS_WRITEBLK OR IOSM_NOFORMAT
149     1093 3     ELSE
150     1094 3     IO_FUNC = IOS_WRITEPBLK;
151     1095 3
152     1096 3     STR$ANALYZE_SDESC_R1 (.FUNC_DESC; SIZE, ADDRESS);
```

```
152      1097 3
153      P 1098 3      RETURN_IF_ERROR_ ($QIO {
154      P 1099 3          FUNC      = .IO_FUNC,
155      P 1100 3          CHAN      = SCB[PSM$L_CHANNEL],
156      P 1101 3          IOSB      = IOB[IOB_Q_IOSB],
157      P 1102 3          ASTADR    = WRITE_AST,
158      P 1103 3          ASTPRM    = IOB,
159      P 1104 3          P1        = .ADDRESS,
160      P 1105 3          P2        = .SIZE));
161      1106 2
162      1107 2
163      1108 2      [PSMS$K_START_TASK]:
164      1109 2      BEGIN
165      1110 2          BIND SCB = .SMB_CONTEXT[] : $BBLOCK;
166      1111 2          LOCAL BUFFER : $BBLOCK [8];
167      1112 2          LOCAL DEVBUFSIZ : VOLATILE;
168      1113 2          LOCAL ITEM_LIST : $ITMLST_DECL (ITEMS=2);
169      1114 3
170      P 1115 3      $ITMLST_INIT (ITMLST=ITEM_LIST,
171      P 1116 3          (BUFADR=BUFFER +4, ITMCOD=DVI$_DEVDEPEND OR DVI$C_SECONDARY),
172      P 1117 3          (BUFADR=DEVBUFSIZ, ITMCOD=DVI$_DEVBUFSIZ OR DVI$C_SECONDARY));
173      1118 3
174      P 1119 3      RETURN_IF_ERROR_ ($GETDVIW
175      P 1120 3          (CHAN=.SCB[PSM$L_CHANNEL], ITMLST=ITEM_LIST));
176      1121 3
177      1122 3      ! Set the page length so the driver can compute number of line
178      1123 3          feeds for devices lacking mechanical form feed -- preserve the
179      1124 3          class type and width information.
180      1125 3
181      1126 3          BUFFER [0,0,08,0] = .SCB[PSM$L_DEVICE_CLASS];
182      1127 3          BUFFER [1,0,08,0] = .SCB[PSM$L_DEVICE_TYPE];
183      1128 3          BUFFER [2,0,16,0] = .DEVBUFSIZ;
184      1129 3          BUFFER [7,0,08,0] = .SCB[PSM$L_FORM_LENGTH];
185      1130 3
186      P 1131 3      RETURN_IF_ERROR_ ($QIOW
187      P 1132 3          (FUNC=IOS_SETMODE, CHAN=.SCB[PSM$L_CHANNEL], P1=BUFFER, P2=8));
188      1133 3
189      1134 2      RETURN_SSS_NORMAL;
190      1135 2
191      1136 2
192      1137 2      [PSMS$K_CLOSE]:
193      1138 3      BEGIN
194      1139 3          BIND SCB = .SMB_CONTEXT[] : $BBLOCK;
195      1140 3          RETURN_STOP_PRINTER (SCB[PSM$L_CHANNEL], SCB[PSMS$Q_DEVICE_NAME]);
196      1141 2
197      1142 2
198      1143 2      [PSMS$K_OPEN]:
199      1144 3      BEGIN
200      1145 3          BIND SCB = .SMB_CONTEXT[] : $BBLOCK;
201      1146 3          LOCAL DEVDEPEND : $BBLOCK[8];
202      1147 3
203      P 1148 3      RETURN_IF_ERROR_ (START_PRINTER (
204      P 1149 3          .FUNC_DESC,          | - specified device name
205      P 1150 3          .FUNC_ARG,          | - resultant device status
206      P 1151 3          SCB[PSM$L_CHANNEL], | - channel assigned
207      P 1152 3          SCB[PSM$L_DEVICE_CLASS], | - device class
208      P 1153 3          SCB[PSM$L_DEVICE_TYPE], | - device type
```

```

: 209 P 1154 3 .FUNC DESC,
: 210 P 1155 3 DEVDEPEND
: 211 1156 3 );
: 212 1157 3
: 213 1158 3 IF .SCB[PSMSL_DEVICE_CLASS] EQL DC$_TERM
: 214 1159 3 THEN
: 215 1160 4 BEGIN
: 216 1161 4 IF NOT .DEVDEPEND[TT$V_MECHTAB]
: 217 1162 4 THEN
: 218 1163 4 SCB[PSMSV_EXPAND_TABS] = 1;
: 219 1164 4
: 220 P 1165 4 RETURN_IF_ERROR ($QIO (
: 221 P 1166 4 FUNC = IOS_SETMODE OR IOSM_CTRLYAST,
: 222 P 1167 4 CHAN = .SCB[PSMSL_CHANNEL],
: 223 P 1168 4 P1 = PSMSHANGUP_DISPATCH_TABLE +
: 224 P 1169 4 (.SCB[PSMSL_STREAM_INDEX] *
: 225 P 1170 4 PSMS_HANGUP_DISPATCH_ENTRY),
: 226 1171 4 P2 = 0));
: 227 1172 4 END
: 228 1173 3 ELSE
: 229 1174 3 IF .SCB[PSMSL_DEVICE_TYPE] EQLU DT$_LP11
: 230 1175 4 AND (NOT .DEVDEPEND[PSV_TAB])
: 231 1176 3 THEN
: 232 1177 3 SCB[PSMSV_EXPAND_TABS] = 1;
: 233 1178 3
: 234 1179 3 RETURN SSS_NORMAL;
: 235 1180 2 END;
: 236 1181 2
: 237 1182 2 [PSMSK_CANCEL]:
: 238 1183 3 BEGIN
: 239 1184 3 BIND SCB = .SMB_CONTEXT[] : $BBLOCK;
: 240 1185 3 RETURN_IF_ERROR_ ($CANCEL (CHAN=.SCB[PSMSL_CHANNEL]));
: 241 1186 2 END;
: 242 1187 2
: 243 1188 2 [OTHERWISE]:
: 244 1189 2 RETURN PSMS_FUNNOTSUP;
: 245 1190 2
: 246 1191 2 TES;
: 247 1192 2
: 248 1193 2 PSMS_PENDING
: 249 1194 2
: 250 1195 1 END;

```

```

.TITLE OUTPUT Print Symbiont -- output routines
.IDENT \V04-000\

.EXTRN BASSEDIT, LBR$CLOSE
.EXTRN LBR$GET_RECORD, LBR$INI_CONTROL
.EXTRN LBR$LOOKUP_KEY, LBR$OPEN
.EXTRN LBR$RET_RM5STV, LBR$SET_LOCATE
.EXTRN LIB$TRIM, FILESPEC
.EXTRN LIB$GET_V$M, LIB$FREE_V$M
.EXTRN STR$ANALYZE_SDESC
.EXTRN STR$ANALYZE_SDESC_R1
.EXTRN STR$APPEND, STR$CONCAT
.EXTRN STR$COPY_DX, STR$COPY_R

```

				.EXTRN	STR\$FREE1 DX, STR\$FREE1_DX_R4
				.EXTRN	STR\$GET1 DX, STR\$LEFT
				.EXTRN	STR\$PREFIX, STR\$RIGHT
				.EXTRN	PSMS\$ HANGUP DISPATCH ENTRY
				.EXTRN	PSMS\$ BUFFEROVF, PSMS\$ EOF
				.EXTRN	PSMS\$ ESCAPE, PSMS\$ FLUSH
				.EXTRN	PSMS\$ FUNNOTSUP, PSMS\$ INVITMCOD
				.EXTRN	PSMS\$ INVVMSOSC, PSMS\$ MODNOTFND
				.EXTRN	PSMS\$ NEWPAGE, PSMS\$ NOFILEID
				.EXTRN	PSMS\$ OSCTOOLON, PSMS\$ PENDING
				.EXTRN	PSMS\$ SUSPEND, PSMS\$ TOOMANYLEV
				.EXTRN	SMB\$\$_INVSTMNBR, SMB\$\$_INVSTRLEV
				.EXTRN	SMB\$\$_NOMOREITEMS
				.EXTRN	PSMS\$REPORT, SMB\$SEND_TO_JOBCTL
				.EXTRN	PSMS\$STORE_ERRORS
				.EXTRN	PSMS\$HANGUP DISPATCH TABLE
				.EXTRN	SYSSQIO, SYSSGETDVIQ
				.EXTRN	SYSSQIOW, SYSSCANCEL
				.PSECT	CODE,NOWRT,2
			OFFC 00000	.ENTRY	PSM\$OUTPUT, Save R2,R3,R4,R5,R6,R7,R8,R9,-
					R10,R11
				MOVAB	SYSSQIO, R6
				SUBL2	#40, SP
				MOVL	@FUNCTION, R3
				CMPBL	R3, #9
				BLSSU	4\$
				CMPBL	R3, #10
				BGTRU	4\$
				MOVL	#32, IO_FUNC
				MOVL	@SMB_CONTEXT, R4
				MOVL	20(R4), R2
				BBCC	#2, 12(R2), 1\$
				BISB2	#160, 84(R2)
				MOVZWL	#716, R0
				RET	
				CMPBL	R3, #10
				BNEQ	3\$
				CMPBL	416(R2), #66
				BNEQ	2\$
				MOVZWL	#288, IO_FUNC
				BRB	3\$
				MOVL	#11, IO_FUNC
				MOVL	FUNC_DESC, R0
				JSB	STR\$ANALYZE_SDESC_R1
				CLRQ	-(SP)
				CLRQ	-(SP)
				PUSHL	SIZE
				PUSHL	ADDRESS
				PUSHL	R4
				PUSHAB	WRITE AST
				PUSHAB	12(R4)
				PUSHL	IO_FUNC
				PUSHL	396(R2)
				CLRL	-(SP)
				CALLS	#12, SYSSQIO
OB	56	00000000G	00 9E 00002		
	5E		28 C2 00009		
	53	OC	BC D0 0000C		
	09		53 D1 00010		
			63 1F 00013		
	0A		53 D1 00015		
			5E 1A 00018		
	55		20 D0 0001A		
	54	04	BC D0 0001D		
	52	14	A4 D0 00021		
	A2		02 E5 00025		
	A2	A0	8F 88 0002A		
	50	02CC	8F 3C 0002F		
			04 00034		
	0A		53 D1 00035	1\$:	
			15 12 00038		
00000042	8F	01A0	C2 D1 0003A		
			07 12 00043		
	55	0120	8F 3C 00045		
			03 11 0004A		
	55		0B D0 0004C	2\$:	
	50	10	AC D0 0004F	3\$:	
			00 16 00053		
			7E 7C 00059		
			7E 7C 0005B		
			50 DD 0005D		
			51 DD 0005F		
			54 DD 00061		
			0000V CF 9F 00063		
			OC A4 9F 00067		
			55 DD 0006A		
			018C C2 DD 0006C		
			7E D4 00070		
			66 OC FB 00072		

10	0122	31 00075	BRW	13\$	1108	
	53	D1 00078	CMPL	R3, #16		
52	04	72 12 0007B	BNEQ	6\$		
50		BC D0 0007D	MOVL	@SMB_CONTEXT, R2	1110	
80	000B0004	6E 9E 00081	MOVAB	ITEM_LIST, \$\$ITMBLKPTR	1117	
80	24	8F D0 00084	MOVL	#720900, (\$\$ITMBLKPTR)+		
80		AE 9E 0008B	MOVAB	BUFFER+4, (\$\$ITMBLKPTR)+		
80	00090004	80 D4 0008F	CLRL	(\$\$ITMBLKPTR)+		
80	1C	8F D0 00091	MOVL	#589828, (\$\$ITMBLKPTR)+		
80		AE 9E 00098	MOVAB	DEVBUFFSIZ, (\$\$ITMBLKPTR)+		
		80 7C 0009C	CLRQ	(\$\$ITMBLKPTR)+		
		7E 7C 0009E	CLRQ	-(SP)	1120	
		7E 7C 000A0	CLRQ	-(SP)		
10		AE 9F 000A2	PUSHAB	ITEM_LIST		
		7E D4 000A5	CLRL	-(SP)		
018C		C2 DD 000A7	PUSHL	396(R2)		
		7E D4 000AB	CLRL	-(SP)		
00000000G	00	08 FB 000AD	CALLS	#8, SYSSGETDVIW		
	79	50 E9 000B4	BLBC	STATUS, 8\$		
20	AE	01A0 C2 90 000B7	MOVB	416(R2), BUFFER	1126	
21	AE	01A4 C2 90 000BD	MOVB	420(R2), BUFFER+1	1127	
22	AE	1C AE B0 000C3	MOVW	DEVBUFFSIZ, BUFFER+2	1128	
27	AE	78 A2 90 000C8	MOVB	120(R2), BUFFER+7	1129	
		7E 7C 000CD	CLRQ	-(SP)	1132	
		7E 7C 000CF	CLRQ	-(SP)		
		08 DD 000D1	PUSHL	#8		
	34	AE 9F 000D3	PUSHAB	BUFFER		
	7E	7E 7C 000D6	CLRL	-(SP)		
018C		23 7D 000D8	MOVQ	#35, -(SP)		
		C2 DD 000DB	PUSHL	396(R2)		
00000000G	00	7E D4 000DF	CLRL	-(SP)		
	03	0C FB 000E1	CALLS	#12, SYSSQIOW		
		50 E9 000E8	BLBC	STATUS, 5\$		
		0094 31 000EB	BRW	11\$		
		04 000EE	RET		1134	
	02	53 D1 000EF	CMPL	R3, #2	1137	
		18 12 000F2	BNEQ	7\$		
7E	04	BC 0000004C	8F C1 000F4	ADDL3	#76, @SMB_CONTEXT, -(SP)	1140
7E	04	BC 0000018C	8F C1 000FD	ADDL3	#396, @SMB_CONTEXT, -(SP)	
0000V	CF		02 FB 00106	CALLS	#2, STOP_PRINTER	
			04 0010B	RET		
	04	53 D1 0010C	CMPL	R3, #4	1143	
		75 12 0010F	BNEQ	12\$		
52	04	BC D0 00111	MOVL	@SMB_CONTEXT, R2	1145	
	20	AE 9F 00115	PUSHAB	DEVDEPEND	1156	
	10	AC DD 00118	PUSHL	FUNC_DESC		
	01A4	C2 9F 0011B	PUSHAB	420(R2)		
	01A0	C2 9F 0011F	PUSHAB	416(R2)		
	018C	C2 9F 00123	PUSHAB	396(R2)		
0000V	7E	10 AC 7D 00127	MOVQ	FUNC_DESC, -(SP)		
	CF	07 FB 0012B	CALLS	#7, START_PRINTER		
00000042	7A	50 E9 00130	BLBC	STATUS, 16\$		
	8F	01A0 C2 D1 00133	CMPL	416(R2), #66	1158	
	04	34 12 0013C	BNEQ	10\$		
0C	A2	AE E8 0013E	BLBS	DEVDEPEND+1, 9\$	1161	
		02 88 00142	BISB2	#2, 12(R2)	1163	
		7E 7C 00146	CLRQ	-(SP)	1171	

50	022C	C2 00000000G 00000000G0040	7E 7C 00148 7E D4 0014A 8F C5 0014C 9F 00156 7E 7C 0015D 7E D4 0015F A3 8F 9A 00161 C2 DD 00165 7E D4 00169 OC FB 0016B 50 E8 0016E 04 00171	CLRQ -(SP) CLRL -(SP) MULL3 #PSM\$S_HANGUP_DISPATCH_ENTRY, 556(R2), R0 PUSHAB PSM\$HANGUP_DISPATCH_TABLE[R0] CLRQ -(SP) CLRL -(SP) MOVZBL #163, -(SP) PUSHL 396(R2) CLRL -(SP) CALLS #12, SYS\$QIO BLBS STATUS, 11\$ RET	
04	20 0C	AE A2 50	01 01A4 09 12 00177 05 E0 00179 02 88 0017E 01 D0 00182 04 00185	10\$: CMPL 420(R2), #1 BNEQ 11\$ BBS #5, DEVDEPEND, 11\$ BISB2 #2, 12(R2) MOVL #1, R0 RET	1174
		01 50 00000000G	53 D1 00186 13 12 00189 BC D0 0018B C0 DD 0018F 01 FB 00193 09 50 E8 0019A	12\$: CMPL R3, #1 BNEQ 14\$ MOVL @SMB_CONTEXT, R0 PUSHL 396(R0) CALLS #1, SYS\$CANCEL BLBS STATUS, 15\$ RET	1175 1177 1179 1182 1184 1185
		00 09 50 00000000G 50 00000000G	04 0019D 04 001A5 8F D0 0019E D0 001A6 04 001AD	13\$: MOVL #PSMS_FUNNOTSUP, R0 RET MOVL #PSMS_PENDING, R0 RET	1189 1195
			14\$: 04 001A5 15\$: 04 001AD 16\$: 04 001AD		

: Routine Size: 430 bytes. Routine Base: CODE + 0000

```
1196 1 %SBTTL 'START_PRINTER'  
1197 1 Functional Description:  
1198 1 ?desc  
1199 1  
1200 1 Formal Parameters:  
1201 1 ?desc  
1202 1 Implicit Inputs:  
1203 1 none  
1204 1  
1205 1 Implicit Outputs:  
1206 1 none  
1207 1  
1208 1 Returned Value:  
1209 1 none  
1210 1  
1211 1 Side Effects:  
1212 1 none  
1213 1 --  
1214 1 ROUTINE START_PRINTER ( %SBTTL 'START_PRINTER'  
1215 1 DEVICE_NAME,  
1216 1 DEVICE_STATUS : REF $LONGWORD,  
1217 1 CHANNEL : REF $WORD,  
1218 1 DEVICE_CLASS : REF $LONGWORD,  
1219 1 DEVICE_TYPE : REF $LONGWORD,  
1220 1 OUT_DEVICE_NAME : ,  
1221 1 OUT_DEVDEPEND : REF VECTOR  
1222 1 ) =  
1223 2 BEGIN  
P 1224 2 PARAMETER INDEX (DEVICE_NAME, DEVICE_STATUS, CHANNEL,  
1225 2 DEVICE_CLASS, DEVICE_TYPE, OUT_DEVICE_NAME, OUT_DEVDEPEND);  
1226 2  
1227 2 LOCAL  
1228 2 CHAN : INITIAL (0),  
1229 2 ITEM_LIST : $ITMLST DECL (ITEMS=4),  
1230 2 DEVDEPEND : $BBLOCK[4] VOLATILE,  
1231 2 DEVSTATUS : INITIAL (0),  
1232 2 DEVCLASS : VOLATILE,  
1233 2 DEVTYPE : VOLATILE,  
1234 2 DEVNAMSIZ : INITIAL (0),  
1235 2 DEVNAMBUF : VECTOR [64, BYTE] VOLATILE,  
1236 2 STATUS_1 :  
1237 2 :  
1238 2  
1239 2 MAP  
1240 2 DEVSTATUS : $BBLOCK;  
1241 2  
P 1242 2 $ITMLST INIT (ITMLST=ITEM_LIST,  
P 1243 2 (BUFADR=DEVDEPEND, ITMCOD=DVIS$_DEVDEPEND OR DVIS$_SECONDARY),  
P 1244 2 (BUFADR=DEVCLASS, ITMCOD=DVIS$_DEVCLASS OR DVIS$_SECONDARY),  
P 1245 2 (BUFADR=DEVTYPE, ITMCOD=DVIS$_DEVTYPE OR DVIS$_SECONDARY),  
P 1246 2 (BUFADR=DEVNAMBUF, ITMCOD=DVIS$_FULLDEVNAM OR DVIS$_SECONDARY),  
P 1247 2 BUFSIZ=%ALLOCATION(DEVNAMBUF), RETLEN=DEVNAMSIZ)  
1248 2 );  
1249 2  
1250 2 IF PARAMETER_PRESENT_ (CHANNEL)  
1251 2 THEN  
1252 2 CHANNEL[] = 0;
```

```
309 1253 2
310 1254 2 STATUS_1 = $ALLOC (DEVNAM=.DEVICE_NAME);
311 1255 2 IF .STATUS_1 EQL SSS_DEVALLOC
312 1256 2 THEN
313 1257 2     STATUS_1 = SSS_DEVALLOC;
314 1258 2 IF NOT .STATUS_1
315 1259 2 THEN
316 1260 2     RETURN .STATUS_1;
317 1261 2
318 1262 2 RETURN_IF_ERROR_ ($ASSIGN (DEVNAM=.DEVICE_NAME, CHAN=CHAN));
319 1263 2
320 1264 2 IF PARAMETER_PRESENT_ (CHANNEL)
321 1265 2 THEN
322 1266 2     CHANNEL[] = .CHAN;
323 1267 2
324 1268 2 RETURN_IF_ERROR_ ($GETDVIW (CHAN=.CHAN, ITMLST=ITEM_LIST));
325 1269 2
326 1270 2 IF PARAMETER_PRESENT_ (OUT_DEVDEPEND)
327 1271 2 THEN
328 1272 2     OUT_DEVDEPEND[0] = .DEVDEPEND;
329 1273 2
330 1274 2 SELECTONE .DEVCLASS OF
331 1275 2     SET
332 1276 2
333 1277 2     [DC$ LP]:
334 1278 2         IF .DEVDEPEND[LP$V_LOWER]
335 1279 2             THEN
336 1280 2                 DEVSTATUS[SMBMSG$V_LOWERCASE] = 1;
337 1281 2
338 1282 2     [DC$ TERM]:
339 1283 3         BEGIN
340 1284 3             DEVSTATUS[SMBMSG$V_TERMINAL] = 1;
341 1285 3             IF .DEVDEPEND[TT$V_MODEM]
342 1286 3             THEN
343 1287 3                 IF .DEVDEPEND[TT$V_REMOTE]
344 1288 3                 THEN
345 1289 3                     DEVSTATUS[SMBMSG$V_REMOTE] = 1
346 1290 3                 ELSE
347 1291 3                     RETURN SSS_HANGUP;
348 1292 3                 IF .DEVDEPEND[TT$V_LOWER]
349 1293 3                 THEN
350 1294 3                     DEVSTATUS[SMBMSG$V_LOWERCASE] = 1;
351 1295 2             END;
352 1296 2
353 1297 2     [OTHERWISE]:
354 1298 3         BEGIN
355 1299 3             RETURN_IF_ERROR_ (STOP_PRINTER (CHANNEL[], .DEVICE_NAME));
356 1300 3             RETURN '0';           !*! Better message needed.
357 1301 2         END;
358 1302 2
359 1303 2     TES;
360 1304 2
361 1305 2 IF NOT PARAMETER_PRESENT_ (CHANNEL)
362 1306 2 THEN
363 1307 2     RETURN_IF_ERROR_ (STOP_PRINTER (CHANNEL[], .DEVICE_NAME));
364 1308 2
365 1309 2 IF PARAMETER_PRESENT_ (DEVICE_STATUS)
```

```
366 1310 2 THEN
367 1311 2     DEVICE_STATUS[] = .DEVSTATUS;
368 1312 2
369 1313 2 IF PARAMETER_PRESENT_(DEVICE_CLASS)
370 1314 2 THEN
371 1315 2     DEVICE_CLASS[] = .DEVCLASS;
372 1316 2
373 1317 2 IF PARAMETER_PRESENT_(DEVICE_TYPE)
374 1318 2 THEN
375 1319 2     DEVICE_TYPE[] = .DEVTYPE;
376 1320 2
377 1321 2 IF PARAMETER_PRESENT_(OUT_DEVICE_NAME)
378 1322 2 THEN
379 1323 2     RETURN_IF_ERROR_(STR$COPY_R (.OUT_DEVICE_NAME, DEVNAMSIZ, DEVNAMBUF));
380 1324 2
381 1325 2     SSS_NORMAL
382 1326 2
383 1327 1 END:
```

.EXTRN SYSS\$ALLOC, SYSS\$ASSIGN

0004 00000 START\_PRINTER:

5E	FF7C	CE	9E	00002	.WORD	Save R2
		7E	D4	00007	MOVAB	-132(SP), SP
		52	D4	00009	CLRL	CHAN
	04	AE	D4	0000B	CLRL	DEVNAMSIZ
50	54	AE	9E	0000E	MOVAB	ITEM_LIST, \$\$ITMBLKPTR
80	000B0004	8F	D0	00012	MOVL	#720900, (\$\$ITMBLKPTR)+
80	50	AE	9E	00019	MOVAB	DEVDEPEND, (\$\$ITMBLKPTR)+
		80	D4	0001D	CLRL	(\$\$ITMBLKPTR)+
80	00050004	8F	D0	0001F	MOVL	#327684, (\$\$ITMBLKPTR)+
80	4C	AE	9E	00026	MOVAB	DEVCLASS, (\$\$ITMBLKPTR)+
		80	D4	0002A	CLRL	(\$\$ITMBLKPTR)+
80	00070004	8F	D0	0002C	MOVL	#458756, (\$\$ITMBLKPTR)+
80	48	AE	9E	00033	MOVAB	DEVTYPE, (\$\$ITMBLKPTR)+
		80	D4	00037	CLRL	(\$\$ITMBLKPTR)+
80	00E90040	8F	D0	00039	MOVL	#15269952, (\$\$ITMBLKPTR)+
80	08	AE	9E	00040	MOVAB	DEVNAMBUF, (\$\$ITMBLKPTR)+
80	04	AE	9E	00044	MOVAB	DEVNAMSIZ, (\$\$ITMBLKPTR)+
		80	D4	00048	CLRL	(\$\$ITMBLKPTR)+
03		6C	91	0004A	CMPB	(AP), #3
		08	1F	0004D	BLSSU	1\$
	0C	AC	D5	0004F	TSTL	12(AP)
		03	13	00052	BEQL	1\$
	0C	BC	B4	00054	CLRW	@CHANNEL
		7E	7C	00057	1\$:	-(SP)
		7E	7C	00059	CLRQ	-(SP)
	04	AC	DD	0005B	PUSHL	DEVICE_NAME
00000000G	00	05	FB	0005E	CALLS	#5, SYS\$ALLOC
00000641	8F	50	D1	00065	CMPL	STATUS_1, #1601
		05	12	0006C	BNEQ	2\$
50	0840	8F	3C	0006E	MOVZWL	#2112, STATUS_1
35		50	E9	00073	2\$:	BLBC
		7E	7C	00076	CLRQ	STATUS_1, 4\$
						-(SP)

		08	AE	9F 00078	PUSHAB	CHAN		
		04	AC	DD 0007B	PUSHL	DEVICE NAME		
			G4	FB 0007E	CALLS	#4, SY5\$ASSIGN		
			50	E9 00085	BLBC	STATUS, 9\$		
			6C	91 00088	CMPB	(AP), #3		
			09	1F 0008B	BLSSU	3\$		
			0C	AC D5 0008D	TSTL	12(AP)		
			04	13 00090	BEQL	3\$		
			6E	B0 00092	MOVW	CHAN, @CHANNEL		
			7E	7C 00096	3\$:	CLRQ	- (SP)	
			7E	7C 00098	CLRQ	- (SP)		
			64	AE 9F 0009A	PUSHAB	ITEM_LIST		
			7E	D4 0009D	CLRL	- (SP)		
			18	AE DD 0009F	PUSHL	CHAN		
			7E	D4 000A2	CLRL	- (SP)		
			08	FB 000A4	CALLS	#8, SY5\$GETDVIW		
			50	E9 000AB	4\$::	BLBC	STATUS, 12\$	
			07	91 000AE	CMPB	(AP), #7		
			0A	1F 000B1	BLSSU	5\$		
			1C	AC D5 000B3	TSTL	28(AP)		
			05	13 000B6	BEQL	5\$		
			50	AE D0 000B8	MOVL	DEVDEPEND, @OUT_DEVDEPEND		
			4C	AE D0 000BD	5\$::	MOVL	DEVCLASS, R0	
			50	D1 000C1	CMPL	R0, #67		
			22	13 000C8	BEQL	7\$		
			50	D1 000CA	CMPL	R0, #66		
			23	12 000D1	BNEQ	8\$		
			40	8F 88 000D3	BISB2	#64, DEVSTATUS		
			52	AE 05 E1 000D7	BBC	#5, DEVDEPEND+2, 7\$		
			51	AE 05 E1 000DC	BBC	#5, DEVDEPEND+1, 6\$		
			52	04 88 000E1	BISB2	#4, DEVSTATUS		
				04 11 000E4	BRB	7\$		
			50	02CC 8F 3C 000E6	MOVZWL	#716, R0		
				04 000EB	RET			
			50	AE 95 000EC	7\$::	TSTB	DEVDEPEND	
				16 18 000EF	BGEQ	10\$		
			52	01 88 000F1	BISB2	#1, DEVSTATUS		
				11 11 000F4	BRB	10\$		
			04	AC DD 000F6	8\$::	PUSHL	DEVICE NAME	
			0C	AC DD 000F9		PUSHL	CHANNEL	
			02	FB 000FC	CALLS	#2, STOP PRINTER		
			50	E9 00101	9\$::	BLBC	STATUS, T8\$	
			50	D4 001C4	CLRL	R0		
				04 00106	RET			
			03	6C 91 00107	10\$::	CMPB	(AP), #3	
				05 1F 0010A	BLSSU	11\$		
			0C	AC D5 0010C	TSTL	12(AP)		
			0E	12 0010F	BNEQ	13\$		
			04	AC DD 00111	11\$::	PUSHL	DEVICE NAME	
			0C	AC DD 00114		PUSHL	CHANNEL	
			02	FB 00117	CALLS	#2, STOP PRINTER		
			50	E9 0011C	12\$::	BLBC	STATUS, T8\$	
			02	6C 91 0011F	13\$::	CMPB	(AP), #2	
			09	1F 00122	BLSSU	14\$		
			08	AC D5 00124	TSTL	8(AP)		
			04	13 00127	BEQL	14\$		
			08	BC 52 D0 00129	MOVL	DEVSTATUS, @DEVICE_STATUS		

OUTPUT  
V04-000Print Symbiont -- output routines  
START\_PRINTERE 12  
16-Sep-1984 02:21:40  
14-Sep-1984 12:55:12  
VAX-11 Bliss-32 V4.0-742  
[PRTSMB.SRC]OUTPUT.B32:1Page 14  
(4)

SEP

		04	6C 91 0012D	14\$:	CMPB (AP), #4		1313
		10	0A 1F 00130		BLSSU 15\$		
		10 BC	AC D5 00132		TSTL 16(AP)		
		05	05 13 00135		BEQL 15\$		
		10 05	AE D0 00137	15\$:	MOVL DEVCLASS, @DEVICE_CLASS		1315
			6C 91 0013C		(CMPB (AP), #5		1317
		14 BC	0A 1F 0013F		BLSSU 16\$		
		06	AC D5 00141		TSTL 20(AP)		
		14 06	05 13 00144		BEQL 16\$		
		48	AE D0 00146	16\$:	MOVL DEVTYPE, @DEVICE_TYPE		1319
		18	6C 91 0014B		(CMPB (AP), #6		1321
		18	18 1F 0014E		BLSSU 17\$		
		18	AC D5 00150		TSTL 24(AP)		
		13	13 13 00153		BEQL 17\$		
		08	AE 9F 00155		PUSHAB DEVNAMBUF		1323
		08	AE 9F 00158		PUSHAB DEVNAMSIZ		
		18 00	18 AC DD 0015B		PUSHL OUT_DEVICE_NAME		
		03	03 FB 0015E		CALLS #3, STR\$COPY_R		
		03	50 E9 00165		BLBC STATUS, 18\$		
		50	01 D0 00168	17\$:	MOVL #1, R0		1327
			04 0016B	18\$:	RET		

; Routine Size: 364 bytes, Routine Base: CODE + 01AE

```

385 1328 1 %SBTTL 'STOP_PRINTER'
386 1329 1 Functional Description:
387 1330 1 ?desc
388 1331 1
389 1332 1 Formal Parameters:
390 1333 1 ?desc
391 1334 1 Implicit Inputs:
392 1335 1 none
393 1336 1
394 1337 1 Implicit Outputs:
395 1338 1 none
396 1339 1
397 1340 1 Returned Value:
398 1341 1 none
399 1342 1
400 1343 1 Side Effects:
401 1344 1 none
402 1345 1 --
403 1346 1 ROUTINE STOP_PRINTER (
404 1347 1 CHANNEL : REF $WORD,
405 1348 1 DEVICE_NAME
406 1349 1 ) =
407 1350 2 BEGIN
408 1351 2
409 1352 2 LOCAL
410 1353 2 STATUS_1,
411 1354 2 STATUS_2
412 1355 2 :
413 1356 2
414 1357 2 STATUS_1 = $DASSGN (CHAN=.CHANNEL[]);
415 1358 2
416 1359 2 CHANNEL[] = 0;
417 1360 2
418 1361 2 STATUS_2 = $DALLOC (DEVNAM=.DEVICE_NAME);
419 1362 2
420 1363 2 IF NOT .STATUS_1
421 1364 2 THEN
422 1365 2 RETURN .STATUS_1;
423 1366 2
424 1367 2 .STATUS_2
425 1368 2
426 1369 1 END:

```

.EXTRN SYSSDASSGN, SYSSDALLOC

0004 00000 STOP_PRINTER:					
					.WORD
00000000G	7E	04	BC	3C 00002	Save R2
	00		01	FB 00006	@CHANNEL, -(SP)
	52		50	DO 0000D	CALLS #1, SYSSDASSGN
		04	BC	B4 00010	MOVL R0, STATUS_1
			7E	D4 00013	CLRW @CHANNEL
		08	AC	DD 00015	CLRL -(SP)
00000000G	00		02	FB 00018	PUSHL DEVICE NAME
	03		52	E8 0001F	CALLS #2, SYSSDALLOC
					BLBS STATUS_1, 1\$

1346  
1357  
1359  
1361  
1363

OUTPUT  
V04-000

Print Symbiont -- output routines  
STOP\_PRINTER

G 12  
16-Sep-1984 02:21:40    VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:55:12    [PRTSMB.SRC]OUTPUT.B32;1

Page 16  
(5)

50                    52 D0 00022    MOVL    STATUS\_1, R0  
                      04 00025 1\$:    RET

; 1365  
; 1369

; Routine Size: 38 bytes.    Routine Base: CODE + 031A

SEP  
V04

```

428      1370 1 %SBTTL 'WRITE_AST'
429      1371 1 | Functional Description:
430      1372 1 |     ?desc
431      1373 1 | Formal Parameters:
432      1374 1 |     ?desc
433      1375 1 | Implicit Inputs:
434      1376 1 |     none
435      1377 1 | Implicit Outputs:
436      1378 1 |     none
437      1379 1 | Returned Value:
438      1380 1 |     none
439      1381 1 | Side Effects:
440      1382 1 |     none
441      1383 1 | --
442      1384 1 | ROUTINE WRITE_AST (
443      1385 1 |     IOB      : REF $BBLOCK
444      1386 1 |           : NOVALUE =
445      1387 1 | BEGIN
446      1388 1 | LOCAL
447      1389 1 |     FUNCTION_STATUS
448      1390 1 | ;
449      1391 2 FUNCTION_STATUS = .IOB[IOB_W_IO_STATUS];
450      1392 2 PSM$REPORT (IOB, FUNCTION_STATUS);
451      1393 2
452      1394 2
453      1395 2
454      1396 2
455      1397 2
456      1398 2
457      1399 2
458      1400 2
459      1401 1 END;

```

0000 00000 WRITE_AST:					
50	04	AC	D0	00002	.WORD Save nothing
7E	0C	AO	3C	00006	MOVL IOB, R0
		5E	DD	0000A	MOVZWL 12(R0), FUNCTION_STATUS
00000000G	00	04	AC	9F 0000C	PUSHL SP
		02	FB	0000F	PUSHAB IOB
		04	00016		CALLS #2, PSM\$REPORT
					RET

; Routine Size: 23 bytes, Routine Base: CODE + 0340

OUTPUT  
VO4-000

Print Symbiont -- output routines  
WRITE\_AST

I 12  
16-Sep-1984 02:21:40  
14-Sep-1984 12:55:12

VAX-11 Bliss-32 V4.0-742  
[PRTSMB.SRC]OUTPUT.B32;1

Page 18  
(7)

: 461 1402 1 END  
: 462 1403 0 ELUDOM

SE  
VO

#### PSECT SUMMARY

Name	Bytes	Attributes
CODE	855	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

#### Library Statistics

File	----- Symbols -----			Pages Mapped	Processing Time
	Total	Loaded	Percent		
\$_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	76	0	1000	00:01.8

#### COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:OUTPUT/OBJ=OBJ\$:OUTPUT MSRC\$:OUTPUT/UPDATE=(ENH\$:OUTPUT)

Size: 855 code + 0 data bytes  
Run Time: 00:26.6  
Elapsed Time: 00:58.1  
Lines/CPU Min: 3170  
Lexemes/CPU-Min: 32415  
Memory Used: 284 pages  
Compilation Complete

0310 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

